

Multi-Modal Interaction for Robotic Mules

Glenn Taylor, Mike Quist, Matt Lanting, Cory Dunham, Patrick Theisen, Paul Muench

Abstract— Today’s soldier carries on average over 100lbs of gear, which takes its toll on the soldier and on the mission. To help mitigate this problem, the US Department of Defense is researching the use of “robotic mules” to move along with squads and help offload some of the excess weight carried by soldiers. The operator control units (OCUs) for these are typically portable computers with tele-operation or point-and-click interfaces. Instead, the DoD wants heads-up, hands-free methods of interaction that can fit seamlessly into the normal squad interaction patterns. This paper describes our research and prototyping in multi-modal interaction with robotic mules, focused on speech and gesture. We present an analysis of squad interactions to help determine the kind of technology useful for user input recognition. We describe an algorithm for gesture recognition using a 9-axis IMU, results of a formative evaluation, and a prototype multi-modal interface that can be used to command a robotic platform.

Keywords—multi-modal interaction, human-robot interface, gesture, speech

I. INTRODUCTION

Today’s infantry soldier carries in excess of 100lbs of gear on average, including weaponry, ammunition, food, water and, increasingly, batteries to power new technology. All of this weight takes a toll on the soldier’s body and on the mission. To help mitigate some of these problems, the US Department of Defense has begun investing in the development of different “robotic mules” to carry the excess weight of the infantry squad. Some example platforms in this class of vehicles includes Boston Dynamic’s Legged Squad Support System (LS3), Virginia Tech’s (GUSS), and Lockheed Martin’s Squad Mobility Support System (SMSS). These robotic vehicles are meant to move along with the infantry squad, freeing the soldier of excess weight like water and food

While the robotic platforms themselves have received a great deal of research attention, the Operator Control Units (OCUs), have lagged behind. Typical OCUs for these systems include some kind of gamepad for tele-operation with a joystick or, in some cases, menu-driven search-and-click interfaces. These OCUs are often ruggedized laptops or tablets carried by the operator, not only adding weight to the operator (a physical burden), but also demand the user’s attention to keep the robot performing the correct task (a cognitive burden).

To help mitigate both the physical and cognitive burdens, the DoD is interested in heads-up, hands-free interfaces that minimize both the amount weight a soldier carries and the amount of time the soldier has to spend attending to it, allowing the user to focus on the more pertinent job of soldiering.

The research and development of such a heads-up, hands-free interface is the focus of this paper, in particular on natural modes of interaction. Based on how soldiers interact with each other today, speech and gesture are the two most common modalities in a squad, and so are used here as a starting point for human-to-robot interaction. We first review the relevant literature. We then present an analysis of a pre-established military gesture language, identifying a practical set of recognition-oriented dimensions for categorizing the types of gestures found in that language. This analysis, as well as an analysis of the expected use cases of a robotic mule, suggests an approach for automatic recognition of gestures. We present a novel approach for gesture recognition with a 9-axis inertial measurement unit (IMU) and some preliminary recognition results on a small set of gestures. While the bulk of the paper focuses on our gesture research, we also present preliminary results of a speech recognition system that largely mirrors the gesture language. Finally, we briefly describe the integration of these components with a representative ground vehicle.

II. RELATED WORK

Often, gesture and speech are seen as “natural” modes of interaction that would (presumably) make computing systems easier to use. Spoken interfaces have become commodities, with the likes of Apple’s Siri, but recognition in noisy environments (such as in the presence of large robots) is still a challenge. Gesture recognition has only begun to be seen in commercial products in the last few years.

Gesture recognition is a rich research field, covering different technology approaches and use cases. Gestures themselves span a wide range of motions and poses, including body pose, head and eye gaze, hand and arm signals, and fine-grained finger movements. Gesture-based interaction with computers has seen a recent explosion of interest with the availability of inexpensive sensors such as Microsoft Kinect™. Vision-based gesture recognition is widely researched [1] but can be very sensitive to lighting conditions. Gesture recognition using hand-held devices such as Nintendo WiiMote has also been researched widely [2], but typically involves a user holding a device in the hand to make gestures.

Gesture recognition research has had different use cases in mind. In some cases, gestures are used as an interface to games as with the WiiMote; in others, it is used for recognizing letters or shapes written in the air [3]. Of particular relevance are

This work was sponsored by the US Army under contract #W56HZV-13-C-0300

Glenn Taylor, Mike Quist, Matt Lanting, Cory Dunham, and Patrick Theisen are with Soar Technology, Inc. (corresponding author: 734-887-7620; email: glenn@soartech.com; quist@soartech.com; matt.lanting@soartech.com; dunham@soartech.com; patrick.theisen@soartech.com)

Paul Muench is with US Army TARDEC (email: paul.muench.civ@mail.mil)

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 26 FEB 2014		2. REPORT TYPE Journal Article		3. DATES COVERED 08-01-2014 to 18-02-2014	
4. TITLE AND SUBTITLE Multi-Modal Interaction for Robotic Mules				5a. CONTRACT NUMBER W56HZV-13-C-0300	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Glenn Taylor; Mike Quist; Matt Lanting; Cory Dunham; Patrick Theisen				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Soar Technology, Inc.,3600 Green Ct,Ste 600,Ann Arbor,Mi,48105				8. PERFORMING ORGANIZATION REPORT NUMBER ; #24504	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army TARDEC, 6501 East Eleven Mile Rd, Warren, Mi, 48397-5000				10. SPONSOR/MONITOR'S ACRONYM(S) TARDEC	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) #24504	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES Multi-Modal Interaction for Robotic Mules there is also a movie to go along with journal article					
14. ABSTRACT Today's soldier carries on average over 100lbs of gear, which takes its toll on the soldier and on the mission. To help mitigate this problem, the US Department of Defense is researching the use of "robotic mules" to move along with squads and help offload some of the excess weight carried by soldiers. The operator control units (OCUs) for these are typically portable computers with tele-operation or point-and-click interfaces. Instead, the DoD wants heads-up, hands-free methods of interaction that can fit seamlessly into the normal squad interaction patterns. This paper describes our research and prototyping in multi-modal interaction with robotic mules, focused on speech and gesture. We present an analysis of squad interactions to help determine the kind of technology useful for user input recognition. We describe an algorithm for gesture recognition using a 9-axis IMU, results of a formative evaluation, and a prototype multi-modal interface that can be used to command a robotic platform.					
15. SUBJECT TERMS multi-modal interaction, human-robot interface, gesture, speech					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Public Release	18. NUMBER OF PAGES 6	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

systems that use gestures to communicate commands to robotic systems [4]. In some cases, gesture is a primary mode; in others, gestures can be used to supplement other modalities, for example, pointing while speaking [5].

Much of the work in gesture systems has been done in laboratory settings with very controlled lighting conditions, or but a few examples include daytime outdoor uses [4]. In contrast, a robotic mule will operate in wide-ranging light and weather environments, and recognition must be reliable at any time, day or night, rain or shine.

Some work has focused on the development of custom gesture languages to make recognition easier [6]. Alternatively, some attention has been paid to established gesture sets such as American Sign Language [7] or military gestures [1, 8]. Like these, our approach uses gestures from an established language. In this paper, we take a step back to look at the qualities of the prescribed gestures as a way to guide recognition.

III. GESTURE ANALYSIS

A. Operational Setting

To consider how an interface for robotic mules might need to recognize gestures, we need to examine how these gestures would be made in practice. Infantry patrol missions, like those expected a robotic mule would participate in, are conducted around the clock, in various lighting and weather conditions. Squads move through various terrain types that may include all kinds of obstacles and occlusions. The user may be moving around the robotic mule, along with the rest of the squad. The squad may hope to hide their visual, audio and electronic signature in the presence of enemy forces, by using camouflage and minimizing noise, but also by emitting as little electromagnetic signal as possible. The operator's distance from the vehicle may vary widely, depending on the task, threat level, formation, and terrain. All of these factors make for a challenging interaction environment, even for soldiers.

B. Gesture Dimensions and Analysis

For the use case of signaling to a robotic mule, we examined *US Army Field Manual 21-60 Visual Signals* as a source of existing gestures that are used in human-to-human communication in different infantry contexts. (This is trained as doctrine in the US Army, though units may adopt their own unit-specific gestures.) The use of a pre-established gesture set presents some interesting challenges. Because there is an existing language, we do not have to generate one from scratch, which saves time and effort. However, with an existing language, we are at the mercy of the designers of that language. We do not get to craft gestures that might be easily recognizable or distinguishable by a computer, and we must deal with ambiguities or overloading that are present.

Analyzing an existing language is an important first step in determining how to go about recognizing it. There have been many gesture taxonomies developed over the years, mostly focused on *gesture as language*, including the semiotic aspects of gesture, gesture's relation to speech, etc. (See [9] for an overview.) Different gesture qualities can have implications for the kinds of technology that might be used to do the recognition. Since these gestures have been developed

organically over time, presumably they were selected because they are easily seen and made by people. However, the military didn't have in mind robotic systems when they were designed. Here, we look from the practical perspective of ease of recognition by a robot.

We focused our analysis on FM 21-60, in particular, the sections having to do with signals to vehicle drivers, combat formations, and infantry patrols (sections 2-2, 2-4, and 2-5), based on their applicability to robotic mules. If there was a day-night distinction, we only included the daytime gestures, yielding 58 gestures. With this gesture subset in mind, we developed a set of seven dimensions to describe these gestures:

















1. Is the gesture a held pose or does it include movement? (Static versus Dynamic) (also from [10])
2. Does the gesture repeat or is given once? (Continuous versus Discrete)
3. Does the gesture include one arm or both? (1-Arm versus 2-Arm)
4. Does the gesture only use arms or does the hand also convey meaning? (Arm-only versus Hand Articulation)
5. Does the gesture use only hands and arms, or are other parts of the body or equipment referred to? (Only Hand/Arm versus Other Reference)
6. Does the person face the intended recipient or does the person face another direction? (Facing Target versus Other Orientation)
7. Does the gesture happen only in the x-y plane of the body, or does it include z-plane? (X-Y plane versus Z Plane)
8. Is the gesture unique or is the same gesture used in different contexts? (Unique versus Overloaded)

These are dimensions are defined further, and examples given, in Table I, which also includes a breakdown of how gestures fall into these categories. Our categorization was partly subjective based on the diagrams given in FM 21-60; we used two judges to assess for each category and broke ties through discussion.

The different qualities of gestures can have an impact on the kinds of technologies needed to perform recognition. For example, two-arm gestures would require that sensors are able to track both arms. Gestures that include articulated hands require a much finer recognition (with higher resolution), at the level of individual fingers. Gestures referring to other body parts or equipment can be especially complex because the system would need to have a sense of those body parts or equipment to understand the meaning of the gesture. Dynamic gestures require tracking the movement of the body whereas static gesture recognition need only identify a pose. Discrete gestures require a different kind of behavior than continuous gestures on the part of the recognizer (repetitive capture) as well as the robotic platform. Gesture orientation (facing the vehicle or away) has an impact on how to interpret the gesture. Gestures x-y plane alone is less information to worry about than if the gesture also includes information in the z-plane between the gesturer and the receiver. If gestures are overloaded instead of unique, then we also need to understand the context in which they were given to know their meaning.

There are some interesting observations we can make about this categorization. For example, over 70% of the gestures analyzed were dynamic gestures, meaning that motion is an important component that needs to be recognized. However, among the static gestures are the "Halt"-type commands, which

TABLE I. GESTURE DIMENSIONS, EXAMPLES, AND ANALYSIS

Dimension	Examples and Analysis		Dimension	Examples and Analysis	
Static vs Dynamic: A static gesture is recognized by being “held” in position for a time; a Dynamic gesture is recognized by the motion it makes through space and time	Dynamic: 70.2%	Static: 29.8%	Only Hand/Arm vs Other Reference: Some gestures are relative only to the body’s frame; others are relative to another body part such as the head or helmet	Body Frame: 77.2% Other: 22.8%	
					
	Follow	Halt		Freeze	Pace Count
Continuous vs Discrete: A continuous gesture invokes a response while the gesture is repeated; a discrete gesture invokes a response until a new command is given	Continuous: 12.5%	Discrete: 87.5%	Facing target vs Facing other: Some gestures face the target; some gestures are oriented away from the person being communicated with.	Facing Target: 47.4% Other: 52.6%	
					
	Move Forward	Halt		Move Right	Advance/Move out
1-Arm vs 2-Arm: Some gestures are made with one arm, some are made with two	1-arm: 51.8%	2-arm: 48.2%	Gestures in x-y plane only vs Include z-plane: Some gestures are perpendicular to the target; others include information in the space between gesturer and target.	X-Y plane only: 56.1% Z-plane: 43.9%	
					
	Halt	Disregard		Action Right	Follow
Just arms vs Include Hand Articulation: Some gestures do not articulate the hands to convey meaning, others do	Just arms: 63.2%	Hand Articulation: 36.8%	Unique vs Overloaded: Some gestures are unique; others have meaning dependent on the context in which they are given.	Unique: 82.8% Overloaded: 17.2%	
					
	Move Right	Message Acknowledged		Halt	Slow Down / Quick Time

are critical to be able to recognize, especially for robotic vehicles. There is also more overlap among gestures than might be hoped for: the same gesture can mean different things in different contexts. For the example given in the table, the same gesture means *opposite things* in different contexts.

IV. APPROACH

Based on the above analysis, and especially the operational setting, one major design choice was between placing the sensors on the robotic platform (e.g., a stereo camera and microphone) and having a device on the user to capture speech and gesture data. Placing sensors on the vehicle is attractive from a user perspective because the user carries no extra gear. However, a platform-mounted gesture recognizer is susceptible to occlusions or obstacles that might appear between the user and the system. Passive sensors are also highly susceptible to variable lighting conditions, and active sensors (e.g., LIDAR) emit energy that could be detected by adversaries. Field of view (FOV) and range are also both major issues to content with: it is very difficult for the user to know where the FOV and range boundaries are, which makes for a challenging user experience when trying to communicate. The user also has to stand in a particular orientation to the sensor to achieve good recognition. Having a microphone quite a distance away from the user, mounted on a noisy platform, poses large speech recognition challenges.

The alternative of a user-carried device has its own tradeoffs. It increases the user’s weight burden and requires some wireless communication framework, but it is not susceptible to occlusions, FOV, or range concerns that are

major problems with platform-mounted sensors. It also has to be able to capture enough of the body’s motion to be recognized as a gesture. A user-worn microphone also has a much better potential for speech recognition.

Given the operational setting, including the potential for a range of lighting conditions, obscuration, and the user moving anywhere around the vehicle, we chose to experiment with a user-worn smartphone that has a 9-axis IMU to capture gestures and speech input. With this approach, we avoid the occlusion and FOV problems, but we are limited to single-arm gestures with no hand articulation. Theoretically, we can capture data to be able to recognize static and dynamic gestures, discrete and continuous, and capture data in x,y, and z dimensions relative to the user.

A. Gesture Recognition Algorithm

We divided the gesture recognition problem into two phases: first, estimating the position and orientation of the device as a function of time; and second, comparing the resulting “state-space curve” to a library of recorded gestures and identifying the closest suitable match.

1) *Phase 1: Data Fusion.* The raw IMU data was sent from the device at a rate of around 50Hz; each packet contained a timestamp and some subset of the three measured quantities (acceleration, angular velocity, and magnetic field), with measurements taken in a reference frame co-moving and co-rotating with the device (the “body frame”). In the absence of measurement errors, these data can be used to derive the device’s position and orientation relative to a fixed reference

frame (the “lab frame”). We process each data packet as follows:

1. Use the current orientation to obtain the angular velocity, acceleration, and magnetic field in the lab frame.
2. Update the current orientation using the lab-frame angular velocity.
3. Subtract the effect of gravity from the lab-frame acceleration. Update the current velocity using the residual acceleration.
4. Update the current position using the lab-frame velocity.

Measurement noise degrades the quality of the orientation estimates (in step 2), and errors in the orientation will propagate back (in step 1) to degrade the other estimates as well, at an accelerating pace. To minimize this degradation, we calibrated the device to remove sensor bias, and added an orientation-correction step. We performed calibration by instructing the user to hold the device stationary for a few seconds in each of several orientations (face up, face down, vertical). During the calibration period, the gyroscope readings were expected to be zero, while the acceleration and magnetic field readings were expected to be constant (in the lab frame) vectors. By appropriately rotating and averaging the measurements, we determined the gyroscope bias vector, the gravitational acceleration, the accelerometer bias vector, and the background magnetic field. These calibration settings were used to correct the raw measurements from the device prior to further analysis. Ideally, one would have to calibrate a given device only once. However, the environment in which the device is used may have an impact on this; for instance, the local magnetic field will differ from location to location.

To prevent orientation errors from accumulating, we included an additional correction step, which leveraged the measured magnetic field. Since the gravitational acceleration and magnetic field vectors in the lab frame are constant, we defined at each point in time the “reference orientation” as the orientation that best aligns the measured (body-frame) acceleration and magnetic field with the predetermined lab-frame values. This orientation is not exactly correct, since the acceleration has a residual component beyond gravity (when the user is accelerating the device), but it gives some indication of the correct orientation. We incorporate it in a fifth step:

5. Adjust the current orientation in the direction of the reference orientation; make a small adjustment when the estimated residual acceleration is large, and vice versa.

The overall data fusion process, which includes the bias correction, compass-based orientation correction, and the usual frame conversion and integration steps, is shown in Fig. 1.

The result of this process is a stream of position and orientation estimates, as shown in Fig. 2 for several distinct gestures. While the orientation estimates are not expected to drift, the position and velocity estimates will diverge from their true values over time, due to the lack of a position correction analogous to step 5. This is already visible in Fig. 2, in that the endpoints of a gesture that returns to its starting position are estimated by the system to be as much as a meter apart.

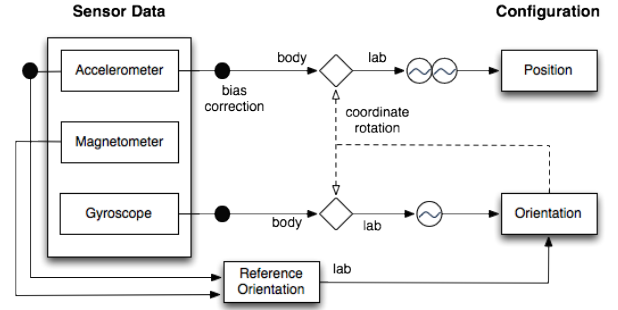


Figure 1. Depiction of the fusion algorithm

2) *Phase 2: Classification:* Having captured a state-space curve, we then want to compare it to a set of pre-recorded and -labeled exemplars (a “gesture library”), and find the type of gesture it most closely resembles. The gesture-to-gesture comparison itself is fairly straightforward, but has a few subtleties worth discussing. When comparing two gestures, we want to use a distance measure that only depends on the essential shape of the gestures. In particular, we want to ignore differences between their overall locations, their absolute headings, and, to some extent, their speeds.

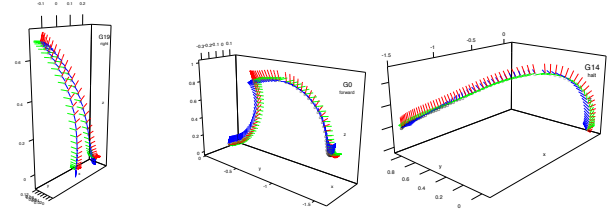


Figure 2. IMU position data for three gestures: Turn-Right, Forward, Stop

We normalized each gesture by (1) translating its initial position to the origin, (2) rotating it around the z-axis so that its most horizontal measurement pointed north, and (3) subtracting a constant-acceleration curve in order to force the final velocity to be zero. Step (3) is intended to correct for the velocity drift discussed in the previous subsection. (The gestures we want to recognize are unambiguous with respect to (2), and do not lose information due to (3); but some gestures would require qualitatively different normalization.) To find the distance between two normalized gestures, ignoring speed differences, we used the Dynamic Time Warping (DTW) algorithm, as described in [11]. DTW finds the optimal pairing between two discretized curves in a metric space, producing a measure of distance that is invariant under an arbitrary rescaling of time (for either curve). In our case, the metric space consists of <orientation, position> pairs; for the distance between two such points, we used:

$$d(\langle o_1, p_1 \rangle, \langle o_2, p_2 \rangle) = \sqrt{d_{\text{rot}}^2(o_1, o_2) + A \cdot d_{\text{euc}}^2(p_1, p_2)}.$$

Here d_{rot} is the usual rotation distance (in radians), d_{euc} is the Euclidean distance (in meters), and A is a conversion factor that we also used to reduce the importance of the (less reliable) position estimate. (We used $A=0.25/\text{m}^2$.) Finally, we added to the total DTW distance an additional term proportional to the rotation distance between the final points of the two gestures.

For our preliminary implementation, we recorded only a dozen or so exemplars for each gesture type. Before doing any recognition, we computed the diameter (i.e., the largest pairwise distance) of each class of exemplars, and used this as an indicator of how close a “good gesture” should be to the exemplars in a given class. To recognize a new gesture, we computed the distance between the gesture and each library gesture, dividing each distance by the library gesture’s class size. (With such a small gesture library, this was computationally feasible; but if there were many more exemplars, we would need to develop a more efficient testing strategy.) The gesture was assigned the class of the best-matching library gesture according to this measure. The result was given a high confidence (measure less than 0.8), low confidence (0.8-1.2), or no confidence (greater than 1.2).

B. Preliminary Evaluation

1) *Command Language*: As a starting point for this work, we chose a subset of the gestures defined in FM 21-60 as a language to control a robot – a mix of infantry patrol gestures and vehicle maneuver gestures: follow me, move forward, stop, turn-left, and turn-right. These are depicted in Fig. 3. These include a mixture of static and dynamic gestures, those that have information in the x-y plane as well as the z-plane, as well as both discrete and continuous gestures. In most cases, the user is facing the platform, but does not have to be for follow-me. We only included gestures that used arm motions – no hand articulation, and no reference to other equipment or body parts.



Figure 3. Implemented Gestures: Follow Me, Move Forward, Stop, Turn Left, Turn Right

Because we chose to use just a single IMU sensor on the user, we could only track a single arm, we were limited to single-arm gestures (or those that had two arms making the same gesture). Because of this, we could not implement a standard *turn-left*, which (doctrinally) is similar to the *turn-right* gesture, but using the opposite arm. Instead, we chose a single-arm mirror turn-right. This is not completely ideal (our *turn-left* overloads the already existing *slow-down* gesture) but some kind of adaptation was required for the single sensor and we didn’t want to invent a completely new gesture.

Voice commands covered the gesture commands and added three more: Move Backward, “Robot” (take control) and “We’re done” (release control).

2) *Gesture Evaluation Procedure and Results*: We recruited 13 users (11 male, 2 female) to test the recognizer, consisting primarily of software engineers. None had prior experience using the system, but most had exposure to the concept. The capture device was a Motorola Moto X smartphone with an on-board 9-axis IMU; data was sent from the phone to a laptop for recognition. The recognizer used a

pre-build gesture library to classify user inputs. The user held the phone in his or her right hand to perform the gesture. All the data was collected in the same location using a single calibration. Each was given an orientation to the task, and had each gesture demonstrated for them. Participants then practiced for a few minutes in front of a display that showed them the recognition result as feedback, until they felt comfortable with the gestures. After this, they were asked to repeat each gesture at least 25 times (first forward, then left, etc.) without the aid of feedback, but with visual depictions in front of them to remind them of the gesture forms. The gesture library against which the user inputs were tested had been built previously.

TABLE II. TABLE 1: GESTURE RECOGNITION CONFUSION MATRIX

Output Input	Forward	Left	Right	Stop	Follow	None	N	Acc. %
Forward	97%	0%	0%	1%	0%	2%	327	97.2%
Left	0%	59%	3%	38%	0%	0%	334	58.7%
Right	0%	0%	98%	1%	0%	0%	325	98.5%
Stop	2%	0%	0%	98%	0%	0%	368	98.1%
Follow	3%	0%	0%	3%	92%	1%	334	92.2%
							1688	88.9%

The confusion matrix of the data is shown in Table II. In most cases, the recognizer performs better when the gesture is performed in a snappier way; more nonchalant movements yield lower performance. This makes sense because the algorithm currently looks for an accelerometer impulse as the trigger to the gesture. *Turn-left* was consistently the most difficult to recognize, and in doing the data collection, we found that it was very sensitive to the orientation of the phone at the start of the gesture. If the user held it face down, then we got very good recognition; if it was held somewhat obliquely, recognition rates dropped. For some users, holding the phone flat with the arm outstretched parallel to the body seems to be difficult ergonomically. A few participants adjusted by starting with their arm slightly out of parallel with their bodies, and in these cases recognition of *turn-left* rose to 78% (N=100, 4 users). Other gestures were not as sensitive to this starting position. More diverse training data for this gesture may help improve these results.

3) *Speech Evaluation Procedure and Results*. We recruited 5 users (2 male, 3 female) to test the speech recognizer, again with the population consisting of mostly software engineers. We used the SPEAR® speech recognizer from Think-a-Move. The audio data was captured on the same smartphone as above using with a custom user interface that included a push-to-talk button to delimit each utterance. We used only the smartphone’s on-board microphone rather than special recording equipment. The smartphone was held roughly 30cm (12 inches) from the user’s mouth. The audio was sent over wi-fi to a Windows7 laptop to perform the speech recognition. For this data collection, each was shown a list of 8 commands to speak, and given the chance to practice a few times. They had the list of commands in front of them during the entire

trial. The participants were asked to repeat each at least 10 times. During all trials, a recording of a large robot was playing to provide background noise. The noise is similar to being next to a motorcycle that occasionally revs its engine, ranging from roughly 75-95 decibels. Results are given in Table III. For space, we just include aggregate results per command. A total of 520 utterances were spoken. There were only 5 instances of confusion with other commands; most of the errors were non-recognition (37 instances).

TABLE III. TABLE 2: SPEECH RECOGNITION RESULTS

Input	Accuracy
"Forward"	90.8%
"Turn left"	98.5%
"Turn right"	100.0%
"Stop"	78.5%
"Follow"	93.8%
"Move Backward"	96.9%
"Robot" (take control)	95.4%
"We're done" (release control)	80.0%
Average Accuracy	91.7%

V. PROTOTYPE IMPLEMENTATION

Using the gesture and speech recognizers described above, we integrated with an rGator, a robotic version of the John Deer Gator. Prior to this test, we had added custom hardware and software for driving and steering the vehicle as well as stereo cameras to perform user tracking and following behaviors. The only other sensors involved for this test were those on the smartphone strapped to the user's wrist: the microphone and the IMU. We used push-to-talk to delimit audio. The CPUs performing recognition were mounted on the rGator itself, and the smartphone communicated user data over wifi. The gesture and speech accuracy outdoors with the robot was generally better than with the study results reported above, likely because our demonstration user had more practice than any of the study participants, but we did not collect data as much outdoors to compare directly.



Figure 4. Figure 1: Demonstration snapshots: turn-right gesture (left pane) and stop gesture (right pane)

VI. CONCLUSIONS AND FUTURE WORK

We have described a prototype system for heads-up, hands-free speech and gesture-based interaction with robotic platforms. The current form factor is a smartphone attached to a user's wrist to capture both speech and gesture data, which is sent to a remote laptop for processing and commanding the vehicle. Smaller form factors are possible, as long as they allow both speech and gesture capture. Speech recognition uses Think-a-Move's SPEAR® speech recognizer. Gesture recognition is based on a 9-axis IMU using a custom recognition algorithm. We have demonstrated this system with

a surrogate robotic mule, in which a user is able to speak or gesture to the robot. The advantage of this gesture recognition approach is that recognition is not susceptible to user position, orientation, lighting, or occlusions. The recognition rates we have seen so far in a quick prototype are very promising. We have also described a practical taxonomy of gestures that has aided us in how we approached gesture recognition.

Our future work includes expanding the gesture and speech vocabulary and providing feedback to the user. We have more work to do on recognition accuracy to be useful in the field, and we'd like to explore whether using a more deliberate trigger for gesture recognition would improve accuracy. We also need to explore the sensitivity of gesture recognition to the initial calibration of the phone, in particular to different magnetic field environments. We will also evaluate the robustness of the system in more natural settings, such as soldiers on patrol with a robotic mule.

VII. REFERENCES

- [1] Y. Song, D. Demirdjian, and R. Davis, "Tracking body and hands for gesture recognition: NATOPS aircraft handling signals database," presented at the IEEE International Conference on Automatic Face & Gesture Recognition (FG 2011), 2011.
- [2] T. Schlomer, B. Poppinga, N. Henze, and S. Boll, "Gesture Recognition with a Wii Controller," presented at the Second Int'l Conference on Tangible and Embedded Interaction, Bonn, Germany, 2008.
- [3] A. Schick, D. Morlock, C. Amma, T. Schultz, and R. Steifelhagen, "Vision-based handwriting recognition for unrestricted text input in mid-air," presented at the International Conference on Multimodal Interaction, 2012.
- [4] G. T. Jay, P. Beeson, and O. C. Jenkins, "Beat-based gesture recognition for non-secure, far-range, or obscured perception scenarios," presented at the IJCAI Workshop on Space, Time, and Ambient Intelligence, Barcelona, Spain, 2011.
- [5] R. A. Bolt, "Put-that-there: Voice and gesture at the graphics interface.," *Computer Graphics*, vol. 14, pp. 262-270, 1980.
- [6] S. Shon, J. Beh, C. Yang, D. Han, and H. Ko, "Motion primitives for designing flexible gesture set in Human-Robot Interface," presented at the 11th International Conference on Control, Automation, and Systems, 2011.
- [7] J. Pansare, S. Gawande, and M. Ingle, "Real-time Static Hand Gesture Recognition for American Sign Language in Complex Background," *Journal of Signal and Information Processing*, vol. 3, pp. 364-367, 2012.
- [8] Beach.G. and C. J. Cohen, "Recognition of computer-based human gestures for device control and interacting with virtual worlds," U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, VA2000.
- [9] A. Kendon, *Gesture: Visible Action as Utterance*. Cambridge, UK: Cambridge University Press, 2004.
- [10] M. Karam and M. C. Schraefel, "A Taxonomy of Gestures in Human Computer Interactions," *ACM Transactions on Computer-Human Interactions*, 2005.
- [11] S. Salvador and P. Chan, "Toward accurate dynamic time wrapping in linear time and space.," *Intelligent Data Analysis*, vol. 11, pp. 561-580, 2007.